# Advance DSA

## Arrays – 1 (7th Jan 2022)

1. **Kadane’s Algorithm**

Used to find the maximum sum of any contiguous array / sub-array. Important point to note is that the array elements should be continuous in nature.

Helps to traverse the array in O(N) TC and without using any extra auxiliary space.

Problems based

1. [**Important**] Given an array, find the maximum sum of contiguous elements. *[Amazon, Directi, Google, LinkedIn].*

Follow-up question is to return the start and end indexes along with the maximum sum. i.e. (start, end) and max.

1. Given an array of integers, A of length N, find out the maximum sum sub-array of non-negative numbers from A.

The sub-array should be contiguous i.e., a sub-array created by choosing the second and fourth element and skipping the third element is invalid.

1. **Marking the presence of the numbers using the same array [*Important – Revise*]**

Reason we are able to use the original array to mark the presence of the numbers is because we are given the range and we are only concerned about the numbers from [1, N] and we also know the size of the array is also N. Hence, we can use the indices to mark the elements presence. If the elements are negative or the range varies than this approach can not be used. We are able to achieve the O(N) TC and without even using any extra space like HashSet.

Problems based

1. Given an array A, find the smallest positive number that is not present in the array. *[Google].*
2. **Preprocessing Data**

Preprocess the given input data to achieve the final solution. It also helps to reduce the overall time complexity.

Problems based

1. Rainwater trapping problem.

Follow up question could you come up with a solution without using any extra auxiliary space? [*Facebook, Google, GS*]